# Pay Today for a Rainy Day: Improved Approximation Algorithms for Demand-Robust Min-Cut and Shortest Path Problems

Daniel Golovin[1,⋆], Vineet Goyal[2,⋆⋆], and R. Ravi[2,⋆⋆]

[1] Computer Science Department, Carnegie Mellon University, Pittsburgh PA 15213, USA
dgolovin@cs.cmu.edu
[2] Tepper School of Business, Carnegie Mellon University, Pittsburgh PA 15213, USA
{vgoyal, ravi}@andrew.cmu.edu

**Abstract.** Demand-robust versions of common optimization problems were recently introduced by Dhamdhere et al. [4] motivated by the worst-case considerations of two-stage stochastic optimization models. We study the demand robust min-cut and shortest path problems, and exploit the nature of the robust objective to give improved approximation factors. Specifically, we give a $(1 + \sqrt{2})$ approximation for robust min-cut and a 7.1 approximation for robust shortest path. Previously, the best approximation factors were $O(\log n)$ for robust min-cut and 16 for robust shortest paths, both due to Dhamdhere et al. [4].

Our main technique can be summarized as follows: We investigate each of the second stage scenarios individually, checking if it can be independently serviced in the second stage within an acceptable cost (namely, a guess of the optimal second stage costs). For the costly scenarios that cannot be serviced in this way ("rainy days"), we show that they can be fully taken care of in a near-optimal first stage solution (i.e., by "paying today").

We also consider "hitting-set" extensions of the robust min-cut and shortest path problems and show that our techniques can be combined with algorithms for Steiner multicut and group Steiner tree problems to give similar approximation guarantees for the hitting-set versions of robust min-cut and shortest path problems respectively.

## 1 Introduction

Robust optimization has been widely studied to deal with the data uncertainty in optimization problems. In a classical optimization problem, all parameters such as costs and demands are assumed to be precisely known. A small change in these parameters can change the optimal solution considerably. As a result, classical optimization is ineffective in those real life applications where robustness to uncertainty is desirable.

Traditional approaches toward robustness have focused on uncertainty in data [3, 12, 13]. In a typical data-robust model, uncertainty is modeled as a finite set of scenarios, where a scenario is a plausible set of values for the data in the model. The objective is to find a feasible solution to the problem which is "good" in all or most scenarios, where various notions of "goodness" have been studied. Some of them include

⋆ Supported in part by NSF ITR grants CCR-0122581 and IIS-0121678
⋆⋆ Supported in part by NSF grant CCF-0430751 and ITR grant CCR-0122581.

B. Durand and W. Thomas (Eds.): STACS 2006, LNCS 3884, pp. 206–217, 2006.

1. Absolute Robustness (min-max): The objective is to find a solution such that the maximum cost over all scenarios is minimized.
2. Robust Deviation (min-max regret): For a given solution, regret in a particular scenario is the difference between cost of this solution in that scenario and the optimal cost in that scenario. In the robust deviation criteria, the objective is to minimize the maximum regret over all scenarios.

More recent attempts at capturing the concept of robust solutions in optimization problems include the work of Rosenblatt and Lee [19] in the facility design problem, and Mulvey *et al.* [14] in mathematical programming. Even more recently, an approach along similar lines has been advocated by Bertsimas *et al.* [1, 2]. Other related works in the data-robust models include heuristics such as branch and bound and surrogate relaxation for efficiently solving the data-robust instances. A research monograph by Kouvelis and Yu [11] summarizes this line of work. An annotated bibliography available online is a good source of references for work in data-robustness [16].

Most of the prior work addresses the problem of robustness under data uncertainty. In this paper, we consider a model which also allows uncertainty in the problem constraints along with the uncertainty in data. We call this model of robustness as *demand-robust* model since it attempts to be robust with respect to problem demands (constraints). Our model is motivated by the recent work in two-stage stochastic programming problems with recourse [7, 5, 9, 17, 20]. In a two-stage stochastic approach, the goal is to find a solution that minimizes the expected cost over all possible scenarios. While the expected value minimization is reasonable in a repeated decision-making framework, one shortcoming of this approach is that it does not sufficiently guard against the worst case over all the possible scenarios. Our demand-robust model for such problems is a natural way to overcome this shortcoming by postulating a model that minimizes this worst-case cost.

Let us introduce the new model with the demand-robust min-cut problem: Given an undirected graph $G = (V, E)$, a root vertex $r$ and costs $c$ on the edges. The uncertainty in demand and costs is modeled as a finite set of scenarios, one of which materializes in the second stage. The $i^{th}$ scenario is a singleton set containing only the node $t_i$. We call the nodes specified by the scenarios *terminals*. An edge costs $c(e)$ in the first stage and $\sigma_i \cdot c(e)$ in the recourse (second) stage if the $i^{th}$ scenario is realized. The problem is to find a set $E_0 \subseteq E$ (edges to be bought in the first stage) and for each scenario $i$, a set $E_i \subseteq E$ (edges to be bought in the recourse stage if scenario $i$ is realized), such that removing $E_0 \cup E_i$ from the graph $G$ disconnects $r$ from the terminal $t_i$. The objective is to minimize the cost function $\max_i\{c(E_0) + \sigma_i \cdot c(E_i)\}$.

Note that in the above model, each scenario has a different requirement (in scenario $i$, $t_i$ is required to be separated from $r$). Such a scenario model allows to handle uncertainty in problem constraints. Another point of difference with the previous data-robust models is that the demand-robust model is two-stage i.e. solution is bought partially in first stage and is then augmented to a feasible solution in the second stage after the uncertainty is realized. However, cost uncertainty in the our demand-robust model is restrictive, as each element becomes costlier by the same factor in a particular scenario in the second stage unlike the data-robust models which handle general cost uncertainties.

### 1.1 Our Contributions

In this paper we consider the shortest path and min-cut problems in the two-stage demand-robust model. In a recent paper Dhamdhere *et al.* [4] introduced the model of demand robustness and gave approximation algorithms for various problems such as min-cut, multicut, shortest path, Steiner tree and facility location in the framework of two-stage demand-robustness. They use rounding techniques recently developed for stochastic optimization problems [7, 8, 17] for many of their results and obtain similar guarantees for the demand-robust versions of the problem. In this paper we crucially exploit and benefit from the structure of the demand-robust problem: *in the second stage, every scenario can pay up to the maximum second stage cost without worsening the solution cost*. This is not true for the stochastic versions where the objective is to minimize the expected cost over all scenarios. At a very high level, the algorithms for the problems considered in this paper are as follows: Guess the maximum second stage cost $C$ in some optimal solution. Using this guess identify scenarios which do not need any first stage "help" i.e. scenarios for which the best solution costs at most a constant times $C$ in the second stage. Such scenarios can be ignored while building the first stage solution. For the remaining scenarios or a subset of them, we build a low-cost first stage solution and prove the constant bounds by a charging argument.

We give the first constant factor approximation for the demand-robust min-cut problem. The charging argument leading to a constant factor argument crucially uses the laminarity of minimum cuts separating a given root node from other terminals. The previous best approximation factor was $O(\log n)$ due to Dhamdhere *et al.* [4].

**Theorem 1.1.** *There is a polynomial time algorithm which gives a* $(1 + \sqrt{2})$ *approximation for the robust min-cut problem.*

For the demand-robust shortest path problem, we give an algorithm with an improved approximation factor of 7.1 as compared to the previous 16-approximation [4].

**Theorem 1.2.** *There is a polynomial time algorithm which gives a 7.1 approximation for the robust shortest path problem.*

Demand-robust shortest path generalizes the Steiner tree problem and is thus **NP**-hard. The complexity of demand-robust min-cut is still open. However, in section 4 we present **NP**-hard generalizations of both problems, together with approximation algorithms for them. In particular, we consider "hitting set" versions of demand-robust min-cut and shortest path problems where each scenario is a set of terminals instead of a single terminal and the requirement is to satisfy at least one terminal (separate from the root for the min-cut problem and connect to the root for the shortest path problem) in each scenario. We obtain approximation algorithms for these "hitting set" variants by relating them to two classical problems, namely Steiner multicut and group Steiner tree.

## 2 Robust Min-Cut

In this section, we present a constant factor approximation for this problem. To motivate our approach, let us consider the robust min-cut problem on trees. Suppose we know the maximum cost that some optimal solution pays in the second stage (say $C$). Any terminal $t_i$ whose min-cut from $r$ costs more than $\frac{C}{\sigma_i}$ should be cut away from $r$ in the first stage. Thus, if we know $C$, we can identify exactly which terminals $U$ should be cut in the first stage. The remaining terminals pay at most $C$ to buy a cut in the second stage. If there are $k$ scenarios, then there are only $k+1$ choices for $C$ that matter, as there are only $k+1$ possible sets that $U$ could be. Though we may not be able to guess $C$, we can try all possible values of $U$ and find the best solution. This algorithm solves the problem exactly on trees.

The algorithm for general graphs has a similar flavor. In a general graph if for any terminal the minimum $r$-$t_i$ cut costs more than $\frac{C}{\sigma_i}$, then we can only infer that the first stage should "help" this terminal i.e. buy some edges from a $r$-$t_i$ cut. In the case of trees, every minimal $r$-$t_i$ cut is a single edge, so the first stage cuts $t_i$ from the root. However, this is not true for general graphs. But we prove that a similar algorithm gives a constant factor approximation using a charging argument. As in the algorithm for trees, we reduce the needed non-determinism by guessing a set of terminals rather than $C$ itself.

**Algorithm for Robust Min-Cut**
$T = \{t_1, t_2, \ldots, t_k\}$ are the terminals, $r \leftarrow$ root .
$\alpha \leftarrow (1 + \sqrt{2})$.

1. For each terminal $t_i$, compute the cost (with respect to $c$) of a minimum $r$-$t_i$ cut, denoted $\mathrm{mcut}(t_i)$.
2. Let $C$ be the maximum second stage cost of some optimal solution. Guess $U := \{t_i : \sigma_i \cdot \mathrm{mcut}(t_i) > \alpha \cdot C\}$.
3. First stage solution: $E_0 \leftarrow$ minimum $r$-$U$ cut.
4. Second stage solution for scenario $i$: $E_i \leftarrow$ any minimum $r$-$t_i$ cut in $G \setminus E_0$

If we relabel the scenarios in decreasing order of $\sigma_i \cdot \mathrm{mcut}(t_i)$, then for every choice of $C$, $U = \emptyset$ or $U = \{t_1, t_2, \ldots, t_j\}$ for some $j \in \{1, 2, \ldots, k\}$. Thus, we need to try only $k+1$ values for $C$. This algorithm runs in $\tilde{O}(k^2 mn)$ time on undirected graphs using the max flow algorithm of Goldberg and Tarjan [6] to find min cuts. The above algorithm $(1+\sqrt{2})$-approximates the robust min-cut problem.

**Proof of Theorem 1.1.** Let $\mathsf{OPT}$ be an optimal solution, let $E_0^*$ be the edge set it buys in stage one, and let $C_0^*$ and $C$ be the amount it pays in the first and second stage, respectively. Let $\alpha$ be a constant to be specified later, and let $U := \{t_i : \sigma_i \cdot \mathrm{mcut}(t_i) > \alpha \cdot C\}$, where $\mathrm{mcut}(t_i)$ is the cost of minimum $r$-$t_i$ cut in $G$ with respect to the cost function $c$. Note that we can handle every terminal $t_i \notin U$ by paying at most $\alpha C$ in the second stage. We will prove that the first stage solution $E_0$, given by the algorithm has cost $c(E_0) \leq (1 + \frac{2}{\alpha-1})C_0^*$. The output solution is thus a $\max\{\alpha, (1+\frac{2}{\alpha-1})\}$-approximation. Setting $\alpha := (1+\sqrt{2})$ then yields the claimed approximation ratio.

To show $c(E_0) \leq (1+\frac{2}{\alpha-1})C_0^*$, we exhibit an $r$-$U$ cut of cost at most $(1+\frac{2}{\alpha-1})C_0^*$. Recall that $\mathsf{OPT}$ buys $E_0^*$ in the first stage. Since $\sigma_i \cdot \mathrm{mcut}(t_i) > C$ for all $t_i \in U$, $E_0$ must "help" each such $t_i$ reduce its second stage cost by a large fraction. The high level idea is as follows: we show how to group terminals of $U$ into equivalence classes such that each edge of $E_0^*$ helps at most two such classes and then cut away each equivalence class from the root using a cut that can be charged to its portion of $E_0^*$.

Formally, let $G = (V, E)$ be our input. Define $G' := (V, E \setminus E_0^*)$. The goal is to construct a low-cost $r$-$U$ cut, $\mathcal{C}$. We include $E_0^*$ in $\mathcal{C}$. This allows us to ignore terminals that $E_0^*$ separates from the root. $U$ is the set of remaining terminals with $\sigma_i \cdot \mathrm{mcut}(t_i) > \alpha \cdot C$. For a terminal $t \in U$, let $Q_t \subset V$ be the $t$ side of some minimum $r$-$t$ cut in $G'$. Lemma 2.1 proves that there exist min cuts such that $\mathcal{F} := \{Q_t : t \in U\}$ is a laminar family (see figure 1). Let $F$ be all the node-maximal elements of $\mathcal{F}$, that is, $F = \{Q \in \mathcal{F} : \forall Q' \in \mathcal{F},$ either $Q' \subseteq Q,$ or $Q' \cap Q = \phi\}$. For $Q \in F$, we say $Q$ *uses* edges $\{(u, v) \in E_0^* \mid Q \cap \{u, v\} \neq \emptyset\}$. Since $\mathcal{F}$ is laminar, all the sets in $F$ are disjoint. It follows that each edge $e \in E_0^*$ can be used by at most two sets of $F$. For each $Q \in F$, we include the edges of $G'$ incident to $Q$ in the cut $\mathcal{C}$, and charge it to the edges of $E_0^*$ it uses as follows:

For a graph $G = (V, E)$ and $Q \subset V$, let $\delta_G(Q) := \{(q, w) | q \in Q, w \in V \setminus Q\} \cap E$ be the boundary of $Q$ in graph $G$. Fix $Q_{t_i} \in F$, let $X = \delta_G(Q_{t_i}) \cap E_0^*$ (edges that $Q_{t_i}$ *uses*) and let $Y = \delta_G(Q_{t_i}) \setminus E_0^*$ (edges of $G'$ incident to $Q_{t_i}$). Since $\delta_G(Q_{t_i})$ is a $r$-$t_i$ cut in $G$,

$$c(\delta_G(Q_{t_i})) = c(X) + c(Y) \geq \mathrm{mcut}(t_i) \tag{2.1}$$

Since $t_i \in U$, $\sigma_i \cdot \mathrm{mcut}(t_i) > \alpha \cdot C$ so with (2.1) we have $(c(X) + c(Y)) > \frac{\alpha \cdot C}{\sigma_i}$. Also, we know that $\mathsf{OPT}$ pays at most $C$ in second stage costs for any scenario which implies $\sigma_i \cdot c(Y) \leq C$. Thus, we have $c(Y) < \frac{1}{\alpha - 1} c(X)$ Thus we can pay for $c(Y)$ by charging it to the cost of $X \subseteq E_0^*$ and incurring an overhead of $1/(\alpha - 1)$ on the charged edges. Since each edge in $E_0^*$ is charged at most twice, the total charge to buy all edges in $\bigcup_{Q \in F} (\delta_G(Q) \setminus E_0^*)$ is at most $\frac{2c(E_0^*)}{\alpha - 1} = \frac{2C_0^*}{\alpha - 1}$. Thus, a minimum $r$-$U$ cut costs at most $(1 + \frac{2}{\alpha - 1}) C_0^*$. □

**Lemma 2.1.** *Let $U$, $Q_t$ be defined as in the proof of Theorem 1.1 Then there exists a minimum $r$-$t$ cut in $G'$ for each terminal $t \in U$ such that $\mathcal{F} := \{Q_t : t \in U\}$ is a laminar family.*

*Proof.* We start with minimally sized sets $Q_t$. That is, for each $t \in U$, $Q_t$ is the $t$ side of a minimum $r$-$t$ cut in $G'$, and every vertex set $Q'$ containing $t$ but not the root such that $|Q'| < |Q_t|$ satisfies $c(\delta_{G'}(Q')) > c(\delta_{G'}(Q_t))$. We claim this family is laminar. Suppose not, then there exists $A := Q_a, B := Q_b,\ a, b \in U$ that violate the laminar property. Thus, $A \cap B \neq \emptyset$, $A \nsubseteq B$, and $B \nsubseteq A$.

Case 1: $a \in A \setminus B,\ b \in B \setminus A$. Let $X := A \cap B$, $A' := A \setminus X$, and $B' := B \setminus X$. Note the cut capacity function of $G'$, defined $f(Q) := c(\delta_{G'}(Q))$, is submodular. We claim that $f(A') \leq f(A)$ or $f(B') \leq f(B)$, contradicting the minimality of $A$ and $B$. Let $c(V_1, V_2)$ denote the sum of costs of edges from $V_1$ to $V_2$ in $G'$, where $V_1, V_2 \subseteq V$. Then

$$f(A) < f(A') \implies c(X, B') + c(X, (V \setminus (A \cup B))) < c(A', X) \tag{2.2}$$
$$f(B) < f(B') \implies c(X, A') + c(X, (V \setminus (A \cup B))) < c(B', X) \tag{2.3}$$

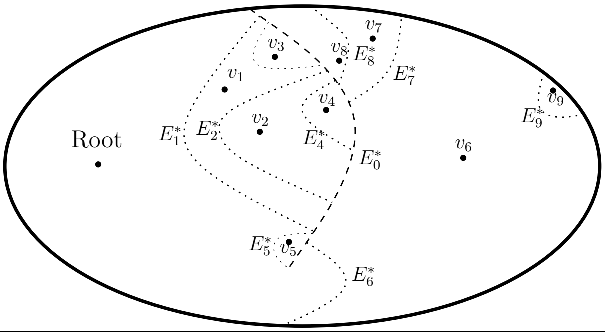


**Fig. 1.** Once the edges bought in the first stage, $E_0^*$, are fixed, there exists an optimal (w.r.t. $E_0^*$) second stage solution $\{E_i^* \mid i = 1, 2, \ldots, k\}$ such that the $t_i$ sides of the cuts $\{E_0^* \cup E_i^*\}$ are a laminar family. Here, the labeled vertices are all terminals, the dashed contour corresponds to $E_0^*$, and the dotted contours correspond to second stage edge sets for various terminals. The node-maximal elements of this family are the terminal side cuts for $v_1, v_5$, and $v_6$.

Adding inequalities (2.2) and (2.3), we get $c(X, (V \setminus (A \cup B))) < 0$, which is clearly impossible.

Case 2: $a \in B$ (equivalently, $b \in A$). Since $A$ and $B$ are terminal sides of min-cuts,

$$\max\{f(A), f(B)\} \leq f(A \cup B) \quad (2.4)$$

$$f(A \cap B) + f(A \cup B) \leq f(A) + f(B) \quad (2.5)$$

where (2.5) follows from submodularity. Inequalities (2.4) and (2.5) together imply $f(A \cap B) \leq \min\{f(A), f(B)\}$. But $f(A \cap B) \leq f(A)$ contradicts the minimality of $A$. □

## 3 Demand-Robust Shortest Path Problem

The problem is defined on a undirected graph $G = (V, E)$ with a root vertex $r$ and cost $c$ on the edges. The $i^{th}$ scenario $S_i$ is a singleton set $\{t_i\}$. An edge $e$ costs $c(e)$ in the first stage and $c_i(e) = \sigma_i \cdot c(e)$ in the $i^{th}$ scenario of the second stage. A solution to the problem is a set of edges $E_0$ to be bought in the first stage and a set $E_i$ in the recourse stage for each scenario $i$. The solution is feasible if $E_0 \cup E_i$ contains a path between $r$ and $t_i$. The cost paid in the $i^{th}$ scenario is $c(E_0) + \sigma_i \cdot c(E_i)$. The objective is to minimize the maximum cost over all scenarios.

The following structural result for the demand-robust shortest path problem can be obtained from a lemma proved in Dhamdhere *et al.* [4].

**Lemma 3.1.** *[4] Given a demand-robust shortest path problem instance on an undirected graph, there exists a solution that costs at most twice the optimum such that the first stage solution is a tree containing the root.*

The above lemma implies that we can restrict our search in the space of solutions where first stage is a tree containing the root and lose only a factor of two. This property is exploited crucially in our algorithm.

**Algorithm for Robust Shortest Path**
Let $C$ be the maximum second stage cost of some fixed connected optimal solution.
$T = \{t_1, t_2, \ldots, t_k\}$ are the terminals, $r \leftarrow$ root, $\alpha \leftarrow 1.775$, $V' \leftarrow \phi$.

1. $V' := \{t_i | \, \mathrm{dist}_c(t_i, r) > \frac{2\alpha \cdot C}{\sigma_i}\}$
2. $\mathcal{B} := \{B_i = B(t_i, \frac{\alpha \cdot C}{\sigma_i}) | \; t_i \in V'\}$, where $B(v, d)$ is a ball of radius $d$ around $v$ with respect to cost $c$. Choose a maximal set $\mathcal{B}_{\mathcal{I}}$ of non-intersecting balls from $\mathcal{B}$ in order of non-decreasing radii.
3. Guess $R^0 := \{t_i | B_i \in \mathcal{B}_{\mathcal{I}}\}$.
4. First stage solution: $E_0 \leftarrow$ The Steiner tree on terminals $R^0 \cup \{r\}$ output by the best approximation algorithm available.
5. Second stage solution for scenario $i$: $E_i \leftarrow$ Shortest path from $t_i$ to the closest node in the tree $E_0$

### 3.1 Algorithm

Lemma 3.1 implies that there is a first stage solution which is a tree containing the root $r$ and it can be extended to a final solution within twice the cost of an optimum solution. We call such a solution as a *connected solution*. Fix an optimal connected solution, say $E_0^*, E_1^*, \ldots, E_k^*$. Let $C$ be the maximum second stage cost paid by this solution over all scenarios, i.e. $C = \max_{i=1}^{k}\{\sigma_i \cdot c(E_i^*)\}$. Therefore, for any scenario $i$, either there is path from $t_i$ to root $r$ in $E_0^*$, or there is a vertex within a distance $\frac{C}{\sigma_i}$ of $t_i$ which is connected to $r$ in $E_0^*$, where distance is with respect to the cost function $c$, denoted $\mathrm{dist}_c(\cdot, \cdot)$. We use this fact to obtain a constant factor approximation for our problem.

The algorithm is as follows: Let $C$ be the maximum second stage cost paid by the connected optimal solution (fixed above) in any scenario. We need to try only $k \cdot n$ possible values of $C$ [1], so we can assume that we have correctly guessed $C$. For each scenario $t_i$, consider a shortest path (say $P_i$) to $r$ with respect to cost $c$. If $c(P_i) \leq \frac{2\alpha \cdot C}{\sigma_i}$, then we can handle scenario $i$ in the second stage with cost only a factor $2\alpha$ more than the optimum. Thus, $t_i$ can be ignored in building the first stage solution. Here $\alpha > 1$ is a constant to be specified later. Let $V' = \{t_i \mid \mathrm{dist}_c(r, t_i) > \frac{2\alpha \cdot C}{\sigma_i}\}$.

For each $t_i \in V'$, let $B_i$ be a ball of radius $\frac{\alpha \cdot C}{\sigma_i}$ around $t_i$. Here, we include internal points of the edges in the ball. We collectively refer to vertices in V and internal points on edges as *points*, $V_P$. Thus, $B_i = \{v \in V_P \mid \mathrm{dist}_c(t_i, v) \leq \frac{\alpha \cdot C}{\sigma_i}\}$.

The algorithm identifies a set of terminals $R^0 \subseteq V'$ to connect to the root in the first stage such that the remaining terminals in $V'$ are close to some terminal in $R^0$ and thus, can be connected to the root in the second stage paying a low-cost.

**Proposition 3.1.** *There exist a set of terminals $R^0 \subseteq V'$ such that:*

1. *For every $t_i, t_j \in R^0$, we have $B_i \cap B_j = \phi$; and*
2. *For every $t_i \in V' \setminus R^0$, there is a representative $\mathrm{rep}(t_i) = t_j \in R^0$ such that $B_i \cap B_j \neq \phi$ and $\frac{\alpha \cdot C}{\sigma_j} \leq \frac{\alpha \cdot C}{\sigma_i}$.*

[1] For each scenario $i$, the second stage solution is a shortest path from $t_i$ to one of the n vertices (possibly $t_i$), so there are at most $k \cdot n$ choices of C.

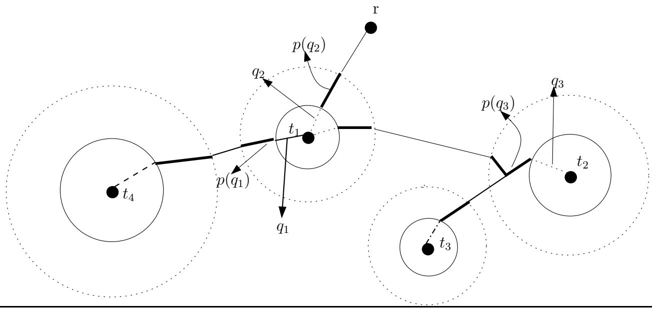


**Fig. 2.** Illustration of first-stage tree computation described in Lemma 3.2. The balls with solid lines denote $B(t_i, \frac{C}{\sigma_i})$, while the balls with dotted lines denote $B(t_i, \frac{\alpha \cdot C}{\sigma_i})$.

*Proof.* Consider terminals in $V'$ in non-decreasing order of the radii $\frac{\alpha \cdot C}{\sigma_t}$ of the corresponding balls $B_t$. If terminal $t_i$ is being examined and $B_i \cap B_j = \phi, \ \forall t_j \in R^0$, then include $t_i$ in $R^0$. If not, then there exists $t_j \in R^0$ such that $B_i \cap B_j \neq \phi$; define $\text{rep}(t_i) = t_j$. Note that $\frac{\alpha \cdot C}{\sigma_j} \leq \frac{\alpha \cdot C}{\sigma_i}$ as the terminals are considered in order of non-decreasing radii of the corresponding balls. □

**The First Stage Tree.** The first stage tree is a Steiner tree on the terminal set $R^0 \cup \{r\}$. However, in order to bound the cost of first stage tree we build the tree in a slightly modified way. For an illustration, refer to Figure 2.

Let $G'$ be a new graph obtained when the balls $B(t_i, \frac{C}{\sigma_i})$ corresponding to every terminal $t_i \in R^0$ are contracted to singleton vertices. We then build a Steiner tree $E_{01}$ in $G'$ with the terminal set as the shrunk nodes corresponding to terminals in $R^0$ and the root vertex $r$. In Figure 2, $E_{01}$ is the union of solid edges and the thick edges. Now, for every shrunk node corresponding to $B(t_i, \frac{C}{\sigma_i})$, we connect each tree edge incident to $B(t_i, \frac{C}{\sigma_i})$ to terminal $t_i$ using a shortest path; these edges are shown as dotted lines in Figure 2 and are denoted by $E_{02}$. Our first stage solution is the Steiner tree $E_0 = E_{01} \cup E_{02}$.

**Lemma 3.2.** *The cost of* $E_0$ *is at most* $\frac{1.55\alpha}{\alpha - 1}$ *times* $c(E_0^*)$*, the first stage cost of the optimal connected solution.*

*Proof.* We know that the optimal first stage tree, $E_0^*$ connects some vertex in the ball $B(t_i, \frac{C}{\sigma_i})$ to the root $r$ for every $t_i \in R^0$, for otherwise the maximum second stage cost of $\mathsf{OPT}$ would be more than $C$. Thus, $E_0^*$ induces a Steiner tree on the shrunk nodes in $G'$. We build a Steiner tree on the shrunk nodes as terminals using the algorithm due to Robins and Zelikovsky [18]. Thus,

$$c(E_{01}) \leq 1.55 \, c(E_0^*) \tag{3.6}$$

Now, consider edges in $E_{02}$. Consider a path $q \in E_{02}$ connecting some edge incident to $B(t_i, \frac{C}{\sigma_i})$ to $t_i$. Since $q$ is the shortest path between its end points, we have $c(q) \leq \frac{C}{\sigma_i}$. Now, consider a path from terminal $t_i$ along $q$ until it reaches $B(t_i, \frac{\alpha \cdot C}{\sigma_i})$ and label the portion between $B(t_i, \frac{C}{\sigma_i})$ and $B(t_i, \frac{\alpha \cdot C}{\sigma_i})$ as $p(q)$. By construction, we have $c(p(q)) \geq \frac{(\alpha-1)\cdot C}{\sigma_i}$, so $c(q) \leq \frac{1}{\alpha-1} \cdot c(p(q))$.

For any two paths $q_1, q_2 \in E_{02}$, the paths $p(q_1)$ and $p(q_2)$ are edge-disjoint. Clearly, if $q_1$ and $q_2$ are incident to distinct terminals of $R^0$, then $p(q_1)$ and $p(q_2)$ are contained in disjoint balls and thus are edge-disjoint. If $q_1$ and $q_2$ are incident to the same terminal, then it is impossible that $p(q_1) \cap p(q_2) \neq \phi$ as $E_{01}$ is a tree on the shrunk graph. Hence, we have

$$\sum_{e \in E_{02}} c(e) = \sum_{q \in E_{02}} c(q) \leq \sum_{q \in E_{02}} \frac{1}{\alpha - 1} \cdot c(p(q)) \leq \sum_{e \in E_{01}} \frac{1}{\alpha - 1} \cdot c(e) \tag{3.7}$$

where the last inequality is due to edge-disjointness of $p(q_1)$ and $p(q_2)$ for any two paths $q_1, q_2 \in E_{02}$. Thus, $c(E_0) = c(E_{01}) + c(E_{02}) \leq c(E_{01}) + \frac{1}{\alpha-1} \cdot c(E_{01}) \leq \frac{1.55\alpha}{\alpha-1} \cdot c(E_0^*)$, where the last inequality follows from (3.6). □

**Second Stage.** The second stage solution for each scenario is quite straightforward. For any terminal $t_i$, $E_i$ is the shortest path from $t_i$ to the closest node in $E_0$.

**Lemma 3.3.** *The maximum second stage cost for any scenario is at most* $2\alpha \cdot C$.

*Proof.* We need to consider the following cases:

1. $t_i \in R^0$: Since the first stage tree $E_0$ connects $t_i$ to $r$, $E_i = \phi$. Thus, $c(E_i) = 0$.
2. $t_i \in V' \setminus R^0$: By Proposition 3.1, there exists a representative terminal $t_j \in R^0$ such that $B_i \cap B_j \neq \phi$ and $\sigma_j \geq \sigma_i$. Therefore, $\text{dist}_c(t_i, t_j) \leq \frac{\alpha \cdot C}{\sigma_i} + \frac{\alpha \cdot C}{\sigma_j} \leq \frac{2\alpha \cdot C}{\sigma_i}$. We know that $t_j$ is connected to $r$ in $E_0$. Thus, the closest node to $t_i$ in the first stage tree is at a distance at most $\frac{2\alpha \cdot C}{\sigma_i}$. Hence, $\sigma_i \cdot c(E_i) \leq 2\alpha \cdot C$.
3. $t_i \notin V'$: Then the shortest path from $t_i$ to $r$ with respect to cost $c$ is at most $\frac{2\alpha \cdot C}{\sigma_i}$. Hence, the closest node to $t_i$ in the first stage tree is at a distance at most $\frac{2\alpha \cdot C}{\sigma_i}$ and $\sigma_i \cdot c(E_i) \leq 2\alpha \cdot C$. □

**Proof of Theorem 1.2.** From Lemma 3.2, we get that $c(E_0) \leq \frac{1.55\alpha}{\alpha-1} c(E_0^*)$. From Lemma 3.3, we get that the second stage cost is at most $2\alpha \cdot C$. Choose $\alpha = \frac{3.55}{2} = 1.775$. Thus, we get $c(E_0) \leq (3.55) \cdot c(E_0^*)$ and $\max_{i=1}^{k}\{\sigma_i \cdot c(E_i)\} \leq (3.55) \cdot C$. From Lemma 3.1 we know that $c(E_0^*) + C \leq 2 \cdot \mathsf{OPT}$, where $\mathsf{OPT}$ is the cost of optimal solution to the robust shortest path instance. Together the previous three inequalities imply $c(E_0) + \max_{i=1}^{k}\{\sigma_i \cdot c(E_i)\} \leq (7.1) \cdot \mathsf{OPT}$ □

## 4 Extensions to Hitting Versions

In this problem, we introduce generalizations of demand-robust min-cut and shortest path problems that are closely related to Steiner multicut and group Steiner tree, respectively. In a Steiner multicut instance, we are given a graph $G = (V, E)$ and $k$ sets of vertices $X_1, X_2, \ldots, X_k$ and our goal is to find the cheapest set of edges $S$ whose removal *separates* each $X_i$, i.e. no $X_i$ lies entirely within one connected component of

$(V, E \setminus S)$. If $\bigcap_{i=1}^{k} X_i \neq \emptyset$, we call the instance *restricted*. In a group Steiner tree instance, we are given a graph $G = (V, E)$, a root $r$, and $k$ sets of vertices $X_1, X_2, \ldots, X_k$ and our goal is to find a minimum cost set of edges $S$ that connects at least one vertex in each $X_i, i = 1, \ldots, k$ to the root $r$. We show how approximation algorithms for these problems can be combined with our techniques to yield approximation algorithms for "hitting versions" of demand-robust min-cut and shortest path problems.

In the hitting version of robust min-cut (resp. shortest path), each scenario $i$ is specified by an inflation factor $\sigma_i$ and a set of nodes $T_i \subset V$ (rather than a single node). A feasible solution is a collection of edge sets $\{E_0, E_1, \ldots, E_k\}$ such that for each scenario $i$, $E_0 \cup E_i$ contains an root-$t$ cut (resp. path) for some $t \in T_i$. The goal is to minimize $c(E_0) + \max_i\{\sigma_i \cdot c(E_i)\}$.

### 4.1 Robust Hitting Cuts

Robust hitting cut is $\Omega(\log k)$-hard, where $k$ is the number of scenarios, even when the graph is a star. In fact, if we restrict ourselves to inputs in which the graph is a star, the root is the center of the star, and $\sigma = \infty$ for all scenarios, then robust hitting cut on these instances is exactly the hitting set problem. In contrast, we can obtain an $O(\log k)$ approximation for robust hitting cut on trees, and $O(\log n \cdot \log k)$ in general using results of Nagarajan and Ravi [15] in conjunction with the following theorem.

**Theorem 4.1.** *If for some class of graphs there is a $\rho$-approximation for Steiner multicut on restricted instances, then for that class of graphs there is a $(\rho+2)$-approximation for robust hitting cut. Conversely, if there is a $\rho$-approximation for robust hitting cut then there is a $\rho$-approximation for Steiner multicut on restricted instances.*

*Algorithm:* Let $\alpha = \frac{1}{2}(\rho + 1 + \sqrt{\rho^2 + 6\rho + 1})$ and let $C$ be the cost that some optimal solution pays in the second stage. For each terminal $t$ in some group, compute the cost of a minimum root-$t$ cut, denoted $\text{mcut}(t)$. Let $\mathcal{T}' := \{T_i : \forall t \in T_i, \sigma_i \cdot \text{mcut}(t) > \alpha \cdot C\}$. Note that there are only $k+1$ possibilities, as in the robust min-cut algorithm. For each terminal set $T_i \in \mathcal{T}'$, separate at least one terminal in $T_i$ from the root in the first stage using an $\rho$-approximation algorithm for Steiner Multicut [10, 15].

**Proof of Theorem 4.1.** We first show that a $\rho$-approximation for robust hitting cut implies a $\rho$-approximation for Steiner multicut on restricted instances. Given a restricted instance of Steiner multicut $(G, X_1, X_2, \ldots, X_k)$ build a robust hitting cut instance as follows: use the same graph and costs, set the root $r$ to be any element of $\bigcap_i X_i$, and create scenarios $T_i = X_i \setminus r$ with $\sigma_i = \infty$ for each $i$. Note that solutions to this instance correspond exactly to Steiner multicuts of the same cost. Thus robust hitting cut generalizes Steiner multicut on restricted instances.

We now show the approximate converse, that a $\rho$-approximation for Steiner multicut on restricted instances implies a $(\rho + 2)$-approximation for robust hitting cut. Let $\mathsf{OPT}$ be an optimal solution, and let $E_0^*$ be the edge set it buys in stage one, and let $C_1$ and $C_2$ be the amount it pays in the first and second stage, respectively. Note we can handle every $T_i \notin \mathcal{T}'$ while paying at most $\alpha \cdot C_2$.

We prove that the first stage edges $E_0 \subset E[G]$ given by our algorithm satisfy all scenarios in $\mathcal{T}'$, and have cost $c(E_0) \leq \rho(1 + \frac{2}{\alpha - 1})C_1$. Thus, the total solution cost is at most $\rho(1+\frac{2}{\alpha-1})C_1 + \alpha \cdot C_2$. Compared to the optimal cost, $C_1 + C_2$, we obtain a $\max\{\alpha, \rho(1+\frac{2}{\alpha-1})\}$-approximation. Setting $\alpha = \frac{1}{2}(\rho + 1 + \sqrt{\rho^2 + 6\rho + 1})$ then yields the claimed $(\rho + 2)$ approximation ratio.

A cut is called a $\mathcal{T}'$*-cut* if it separates at least one terminal in each $T \in \mathcal{T}'$ from the root. There exists a $\mathcal{T}'$-cut of cost at most $(1 + \frac{2}{\alpha-1})C_1$, by the same argument as in the proof of Theorem 1.1. Suppose $\mathsf{OPT}$ cuts away $t_i^*$ when scenario $T_i$ occurs. Then $\mathsf{OPT}$ is also an optimal solution to the robust min-cut instance on the same graph with terminals $\{t_i^* \mid i = 1, 2, \ldots, k\}$ as $k$ scenarios. Since, for all $t \in T$ such that $T \in \mathcal{T}'$, we have $\sigma_t \cdot \mathrm{mcut}(t) > \alpha \cdot C$, we can construct a root-$\{t_i^* \mid i = 1, 2, \ldots, k\}$ cut of cost at most $(1 + \frac{2}{\alpha-1})C_1$. Thus, the cost of an optimal $\mathcal{T}'$-cut is at most $(1 + \frac{2}{\alpha-1})C_1$. Now apply the $\rho$-approximation for Steiner multicut on restricted instances. To build the Steiner multicut instance, we use the same graph and edge costs, and create a groups $X_i = T_i \cup \{\text{root}\}$ for each $T_i \in \mathcal{T}'$. Clearly, the instance is restricted. Note that every solution to this instance is a $\mathcal{T}'$-cut of the same cost, and vice-versa. Thus a $\rho$-approximation for for Steiner multicut on restricted instances yields a $\mathcal{T}'$-cut of cost at most $\rho(1 + \frac{2}{\alpha-1})C_1$. □

**Corollary 4.1.** *There is a polynomial time $O(\log n \cdot \log k)$-approximation algorithm for robust hitting cut on instances with $k$ scenarios and $n$ nodes, and an $O(\log k)$-approximation algorithm for robust hitting cut on trees.*

### 4.2 Robust Hitting Paths

**Theorem 4.2.** *If there is a $\rho$-approximation for group Steiner tree then there is a $2\rho$-approximation for robust hitting path. If there is a $\rho$-approximation for robust hitting path, then there is a $\rho$-approximation for group Steiner tree.*

*Proof.* Note that robust hitting path generalizes group Steiner tree (given a GST instance with graph $G$, root $r$ and groups $X_1, X_2, \ldots, X_k$, use the same graph and root, make each group a scenario, and set $\sigma_i = \infty$ for all scenarios $i$). Thus a $\rho$-approximation for robust hitting path immediately yields a $\rho$-approximation for group Steiner tree.

Now suppose we have an $\rho$-approximation for group Steiner tree. Lemma 3.1 guarantees that there exists a solution $\{E_0, E_1, \ldots, E_k\}$ of cost at most $2\mathsf{OPT}$ whose first stage edges, $E_0$, are a tree containing root $r$.

The algorithm is as follows. Guess $C := \max_i\{\sigma_i c(E_i)\}$. Note that for each scenario $i$ the tree $E_0$ must touch one of the balls in $\{B(t, C/\sigma_i) | t \in T_i\}$, where $B(v, x) := \{u | \operatorname{dist}_c(v, u) \le x\}$. Thus we can construct groups $X_i := \bigcup_{t \in T_i} B(t, C/\sigma_i)$ for each scenario $i$ and use the $\rho$-approximation for group Steiner tree on these groups to obtain a set of edges $E_0'$ to buy in the first stage.

Note that $c(E_0') \le \rho c(E_0)$ and any scenario $i$ has a terminal $t \in T_i$ that is within distance $C/\sigma_i$ of some vertex incident on an edge of tree $E_0'$. We conclude that the total cost is at most $\rho c(E_0) + C \le 2\rho \cdot \mathsf{OPT}$. □

## 5 Conclusion

In this paper we give improved approximation algorithms for robust min-cut and shortest path problems and extend our results to an interesting "hitting-set" variant. It would be interesting to use the techniques introduced in this paper to obtain better approximations for robust minimum multicut and Steiner tree problems. The technique of guessing and pruning crucially uses the fact that each scenario can pay up to the maximum second stage cost without worsening the optimal cost. However, this is not true for the stochastic optimization problems and hence our technique doesn't extend to stochastic versions in a straightforward way. It would be interesting to adapt this idea for stochastic optimization.

## References


1. D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming Series B*, 98:49–71, 2003.
2. D. Bertsimas and M. Sim. The price of robustness. *Operation Research*, 52(2):35–53, 2004.
3. G. B. Dantzig. Linear programming under uncertainty. *Management Sci.*, 1:197–206, 1955.
4. K. Dhamdhere, V. Goyal, R. Ravi, and M. Singh. How to pay, come what may: Approximation algorithms for demand-robust covering problems. In *FOCS*, pages 367–378, 2005.
5. K. Dhamdhere, R. Ravi, and M. Singh. On two-stage stochastic minimum spanning trees. In *IPCO*, pages 321–334, 2005.
6. A. V. Goldberg and R. E. Tarjan. A new approach to the maximum-flow problem. *J. ACM*, 35(4):921–940, 1988.
7. A. Gupta, M. Pál, R. Ravi, and A. Sinha. Boosted sampling: approximation algorithms for stochastic optimization. In *STOC*, pages 417–426, 2004.
8. A. Gupta, R. Ravi, and A. Sinha. An edge in time saves nine: LP rounding approximation algorithms for stochastic network design. In *FOCS*, pages 218–227, 2004.
9. N. Immorlica, D. Karger, M. Minkoff, and V. Mirrokni. On the costs and benefits of procrastination: Approximation algorithms for stochastic combinatorial optimization problems. In *SODA*, pages 684–693, 2004.
10. Philip N. Klein, Serge A. Plotkin, Satish Rao, and Eva Tardos. Approximation algorithms for steiner and directed multicuts. *J. Algorithms*, 22(2):241–269, 1997.
11. P. Kouvelis and G. Yu. *Robust Discrete Optimisation and Its Applications*. Kluwer Academic Publishers, Netherlands, 1997.
12. R. D. Luce and H. Raiffa. *Games and Decision*. Wiley, New York, 1957.
13. J. W. Milnor. Games against nature. In R. M. Thrall, C. H. Coombs, and R. L. Davis, editors, *Decision Processes*. Wiley, New York, 1954.
14. J. M. Mulvey, R. J. Vanderbei, and S. J. Zenios. Robust optimization of large-scale systems. *Operations Research*, 43:264–281, 1995.
15. V. Nagarajan and R. Ravi. Approximation algorithms for requirement cut on graphs. In *APPROX + RANDOM*, pages 209–220, 2005.
16. Y. Nikulin. Robustness in combinatorial optimization and scheduling theory: An annotated bibliography, 2004. `http://www.optimization-online.org/DB_FILE/2004/11/995.pdf`.
17. R. Ravi and A. Sinha. Hedging uncertainty: Approximation algorithms for stochastic optimization problems. In *IPCO*, pages 101–115, 2004.
18. G. Robins and A. Zelikovsky. Improved steiner tree approximation in graphs. In *SODA*, pages 770–779, 2000.
19. M. J. Rosenblatt and H. L. Lee. A robustness approach to facilities design. *International Journal of Production Research*, 25:479–486, 1987.
20. D. Shmoys and C. Swamy. Stochastic optimization is (almost) as easy as deterministic optimization. In *FOCS*, pages 228–237, 2004.